**Generating Random Test Data**

Aby napisać nasze narzędzie do uzgadniania pokwitowań, musimy mieć pewne paragony do pracy, ponieważ testujemy naszą implementację. Oczekujemy, że wpływy będą plikami JSON zawierającymi określone dane i będziemy pisać skrypt, który utworzy dla nas kilka wpływów.

Pracujemy nad systemem wymagającym pewnych ścieżek lokalnych, więc umieśćmy to, co robimy w katalogu pokwitowań:

$ mkdir -p receipts/new

$ cd receipts

Potwierdzenia, które nie zostały uzgodnione, trafią do nowego katalogu, więc już to stworzyliśmy. Stwórzmy plik gen\_receipts.py, aby utworzyć nieuzgodnione pokwitowania, gdy go uruchomimy:

import random

import os

import json

count = int(os.getenv("FILE\_COUNT") or 100)

words = [word.strip() for word in open('/usr/share/dict/words').readlines()]

for identifier in range(count):

amount = random.uniform(1.0, 1000)

content = {

'topic': random.choice(words),

'value': "%.2f" % amount

}

with open(f'./new/receipt-{identifier}.json', 'w') as f:

json.dump(content, f)

Używamy funkcji json.dump, aby upewnić się, że wypisujemy prawidłowy JSON (przeczytamy go później). random.choice pozwala nam wybrać jeden element z iteracji (str, krotka lub lista). Funkcja random.uniform daje nam float między dwoma określonymi granicami. Ten kod pokazuje nam, jak utworzyć zakres, który pobiera numer początkowy i numer końcowy i może być iterowany przez wartości między.

Teraz możemy uruchomić nasz skrypt za pomocą polecenia python3.6:

$ FILE\_COUNT=10 python3.6 gen\_receipts.py

$ ls new/

receipt-0.json receipt-2.json receipt-4.json receipt-6.json receipt-8.json

receipt-1.json receipt-3.json receipt-5.json receipt-7.json receipt-9.json

$ cat new/receipt-0.json

{"topic": "microceratous", "value": "918.67"}